

ABSTRACT

A bytecode execution system and method for increasing the execution speed of invoking and returning from Methods while minimizing the memory footprint required to support this. The system includes a virtual machine with a

5 class loader and an interpreter or, alternatively, a hardware accelerator. Speed and memory enhancements are realized by establishing an activation stack frame template with a set of criteria. Each Method from subject code is examined to determine whether the Method conforms to the criteria of the stack frame template. If the Method conforms, an activation frame for that Method based on

10 the activation stack frame template is created when the Method is invoked. Otherwise, an activation frame based on standard virtual machine criteria is created. An access structure is associated with each Method and a Method routing structure is created for each class. The value of a pointer to the access structure indicates where the Method's implementation resides (bytecode or

15 compiled code) and where the Method's runtime information (stack size, name etc.) resides. The Method routing structure has one or more misaligned pointers to indicate access structures for compiled representations of bytecode and one or more aligned pointers to indicate access structures for bytecode. Invocation bytecodes are rewritten to include an indication of the Method routing structure

20 entry.

T:\CLIENTA\048487\9053\A0283549